Association for Computing Machinery

Austin ACM SIG KDD – Austin's Big Data Machine Learning Group

Advanced Machine Learning with Python

Session 6: Random Forests

Ulind Narang

4/20/2016

Agenda

- Random forests context - history and taxonomy of ML algorithms
- Decision Trees
  - Building a Classification Tree
  - Criteria for Splitting: feature space, test function
  - Structure of Decision Tree
- Random Forests
  - Algorithm
  - Building a random forest
  - Parameters: size, depth, weak learner model, objective function, bagging
- scikit-learn Cookbook – Python code
  - RandomForestClassifier(…)
  - Code review, Demo
  - Tuning
- Applications … discussion
  - Text Classification, Face Detection, Object Detection, Kinect
- References

Sources

- Random forests in the taxonomy of ML algorigthms
- Decision Trees
  - Building a Classification Tree
  - Criteria for Splitting: feature space, test function
  - Structure of Decision Tree
- Random Forests
  - Algorithm
  - Building a random forest
  - Parameters: size, depth, weak learner model,…

> - Criminisi et all. 2011. Decision forests… Foundations and Trends in Computer Graphics and Vision, 7(2-3): 81–227, 2011
> - Lectures by Nando de Freitas. Machine Learning - Random forests. University of British Columbia. 2013

- scikit-learn Cookbook – Python code
  - RandomForestClassifier(…)
  - Code review, Demo
  - Tuning

> - scikit-learn.org + Book

- Applications
  - Text Classification, Face Detection, Object Detection, Kinect

> - Criminisi et. all 2011
> - Nando de Freitas lectures. UBC. 2013

- References

Random Forests – brief history

- 90s: ensembles of learners (generic weak classifiers) yields greater accuracy & generalization – particularly true of high dimensional data in real life.

- Combining the ideas of decision trees and ensemble methods gave rise to decision forests, that is, ensembles of randomly trained decision trees.

- Tree training via randomized partitioning of the feature space and use in forests yields superior generalization to both boosting and pruned trained trees, on some tasks.

- Injecting randomness in the forest by randomly sampling the labeled training data (namely "bagging")

- Techniques for predicting forest test error

- (…classification, regression, density estimation, manifold learning, semi-supervised learning, and active learning…)

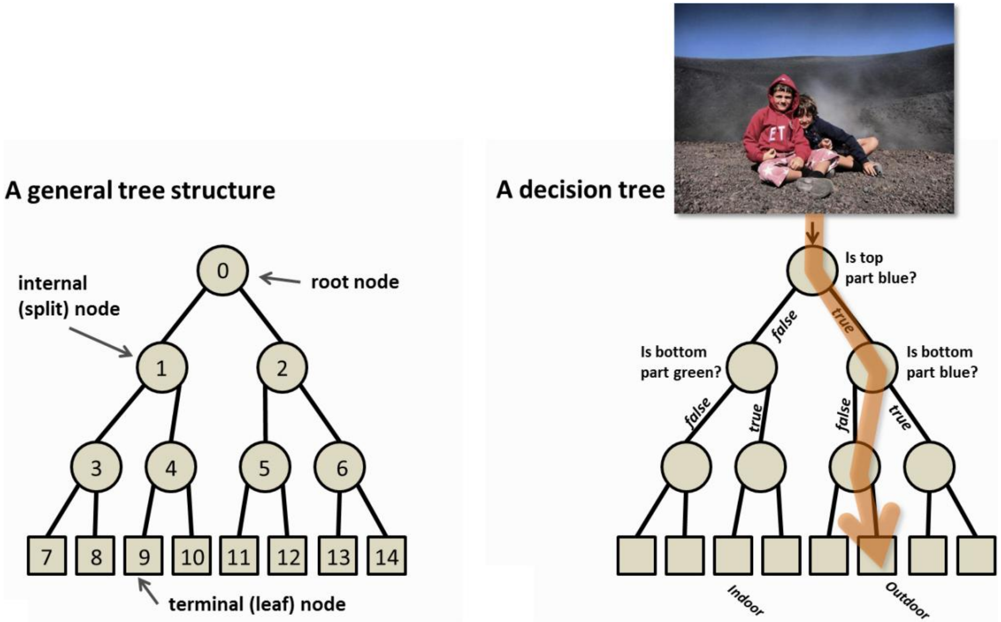
Random Forests – taxonomy

- …Paper by Leo Brieman, 2001
- Supervised Learning
  - Decision Trees
  - Ensemble method
    - Averaging Method
      - Forest of Randomized Trees
      - Bagging Methods
    - Boosting Methods
- …(semi-supervised)
- Unsupervised Learning

Random Forests

DECISION TREES

Decision Trees

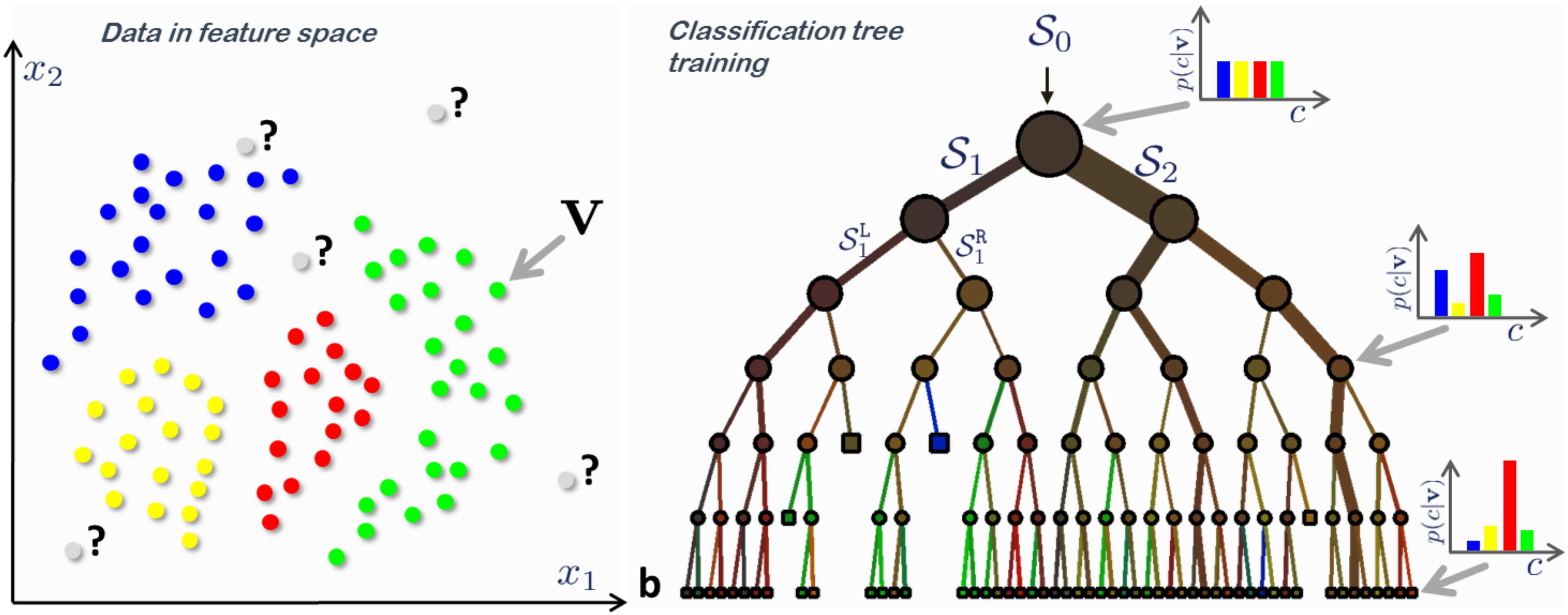


(a) (b)

$$S_j = S_j^{\mathrm{L}} \cup S_j^{\mathrm{R}},$$
$$S_j^{\mathrm{L}} \cap S_j^{\mathrm{R}} = \emptyset,$$

- Tree – as a special graph, n-ary trees
- Decision Tree - set of questions organized in a hierarchical manner and represented graphically as a tree.
- Well functioning tree
  - tests associated with each internal node
  - decision-making predictors associated with each leaf.

Decision Trees – building a classification tree



(a) (b)

- Constructing the tree node-by-node …

$$h(\mathbf{v}, \boldsymbol{\theta}_j) : \mathcal{F} \times \mathcal{T} \rightarrow \{0, 1\},$$

Feature Space

v: a data point denoted by $\mathbf{v} = (x_1, x_2, \ldots, x_d) \in \mathcal{F}$

x_i: represent some attributes of the data point - features

$\mathcal{F}$: Feature space, *d: dimensionality of feature space*

…not all features are 'of interest'. So, extract only a small portion as needed -

$$\phi(\mathbf{v}) = (x_{\phi_1}, x_{\phi_2}, \ldots, x_{\phi_{d'}}) \in \mathcal{F}^{d'} \subset \mathcal{F}$$

Test Function

$$h(\mathbf{v}, \boldsymbol{\theta}_j) : \mathcal{F} \times \mathcal{T} \rightarrow \{0, 1\}$$

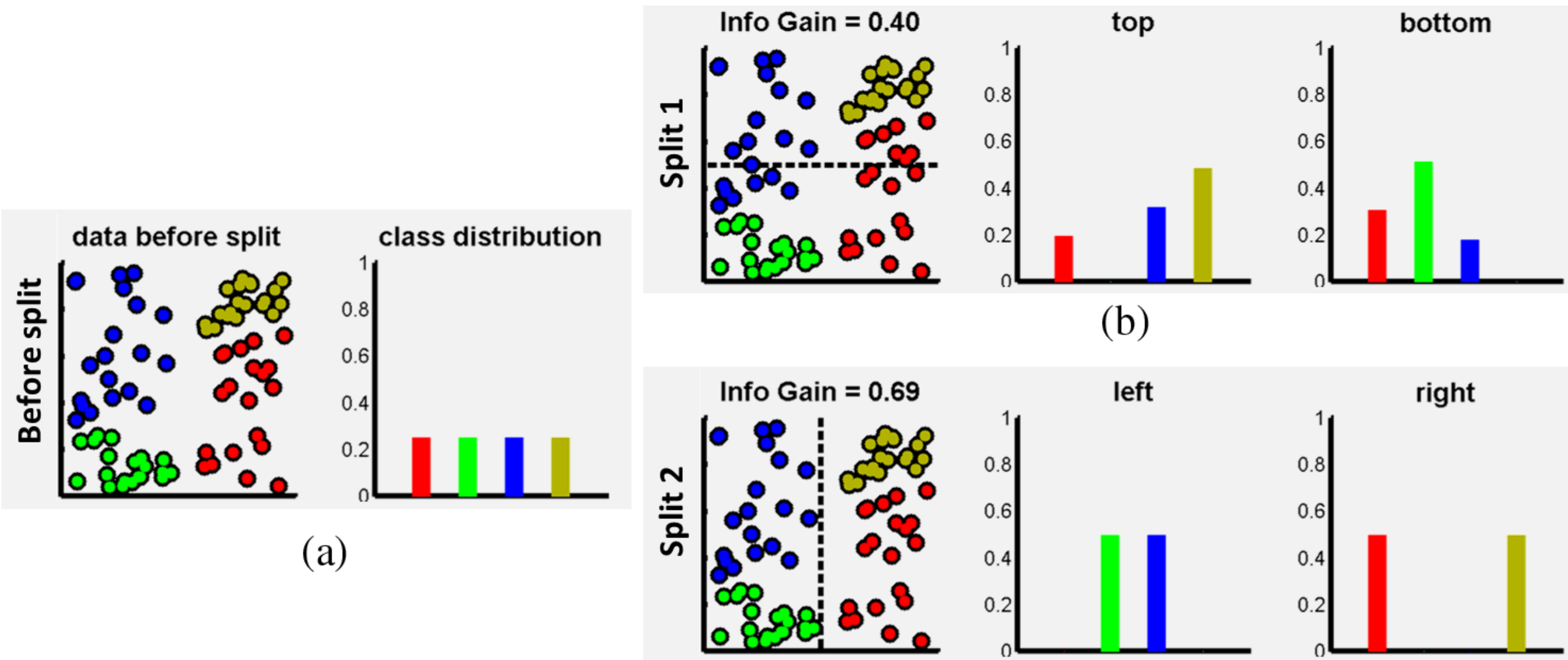
0,1: False and True

$\theta_j \in \mathcal{T}$: parameters of test function at the $j\mathrm{th}$ split node.

Data point v arriving at the split node is sent to its left or right child according to Test Function

Criteria for Splitting: Information Gain

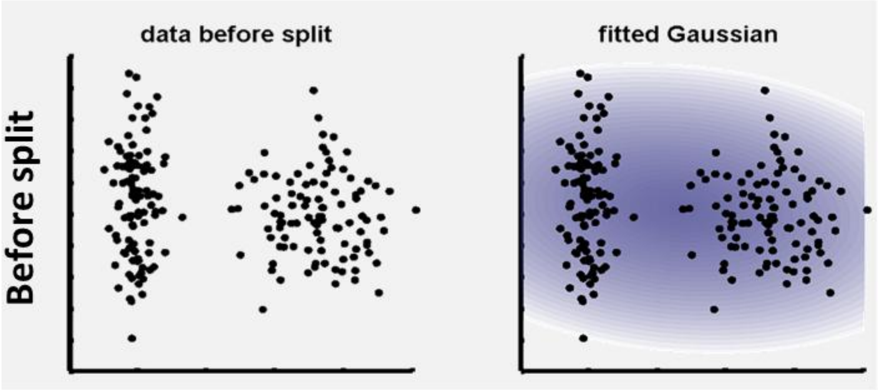
Information Gain for discrete non-parametric distributions



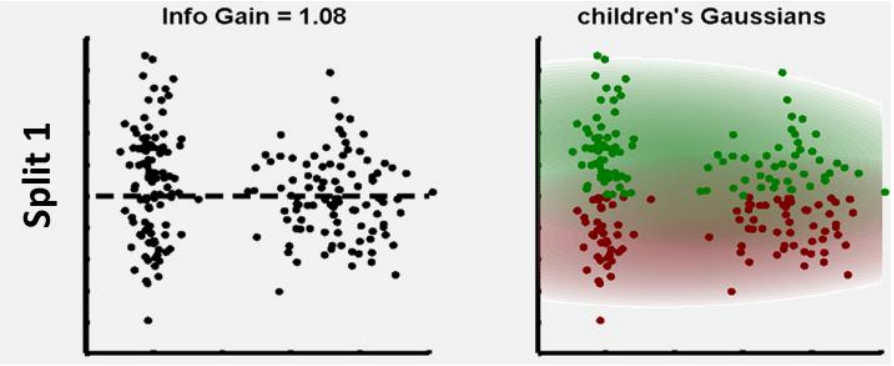
(a)
(b)
(c)

Criteria for Splitting: Information Gain

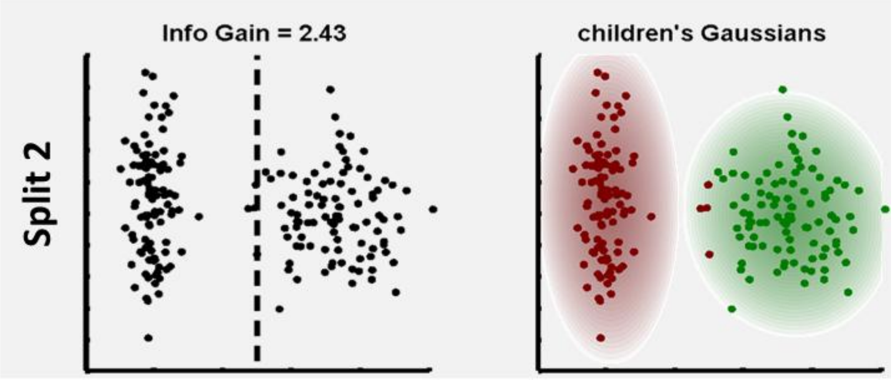
Information Gain for continuous parametric densities



(a)

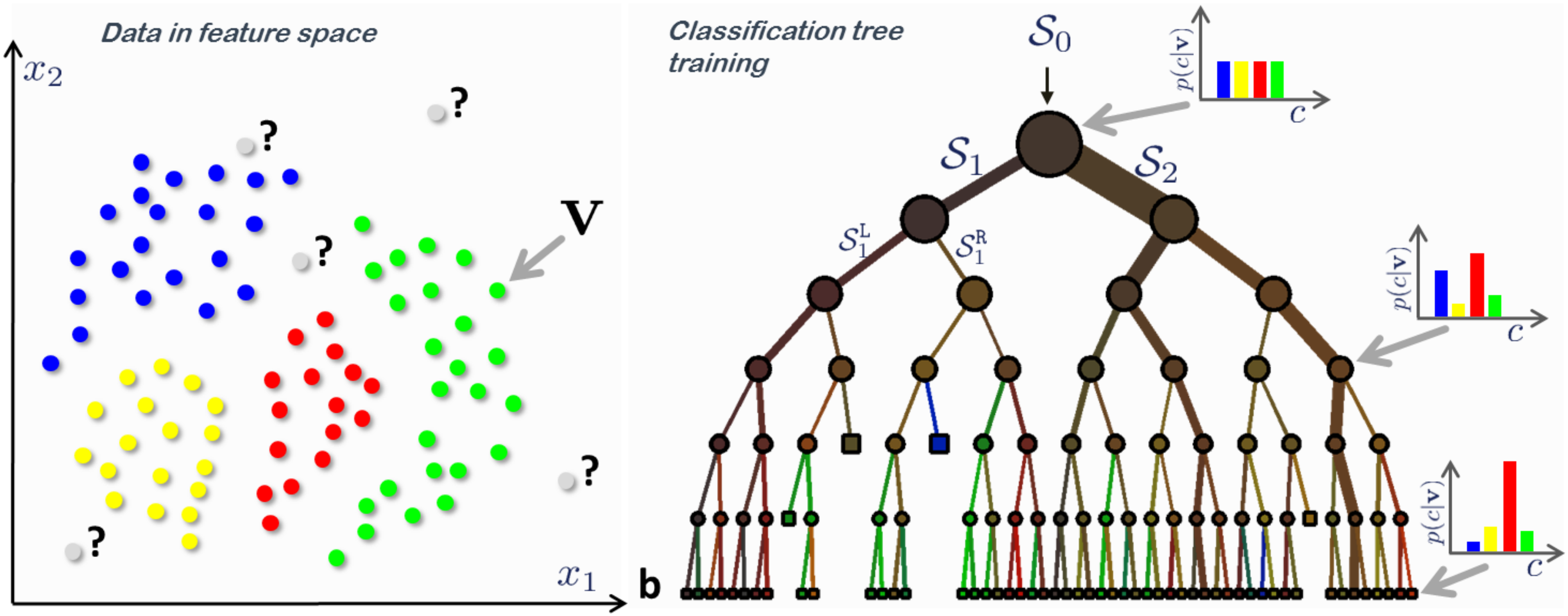


(b)



(c)

Decision Trees → Classification Tree



(a) (b)

- Leaf nodes contain a predictor/estimator associating an output (a.k.a. Class) with input v

Structure of Decision Tree

- Node Split: weak learner model, testing function
  - Gini impurity: purity of node
  - Entropy: Information gain - reduction in entropy

  …leads to automatic creation of decision trees.
- How many branches does a node have?
  - Binary, n-ary
- What is the depth of the tree?
  - When to stop growing branches
    - When maximum depth has been reached
    - Node contains too few training points

Random Forests

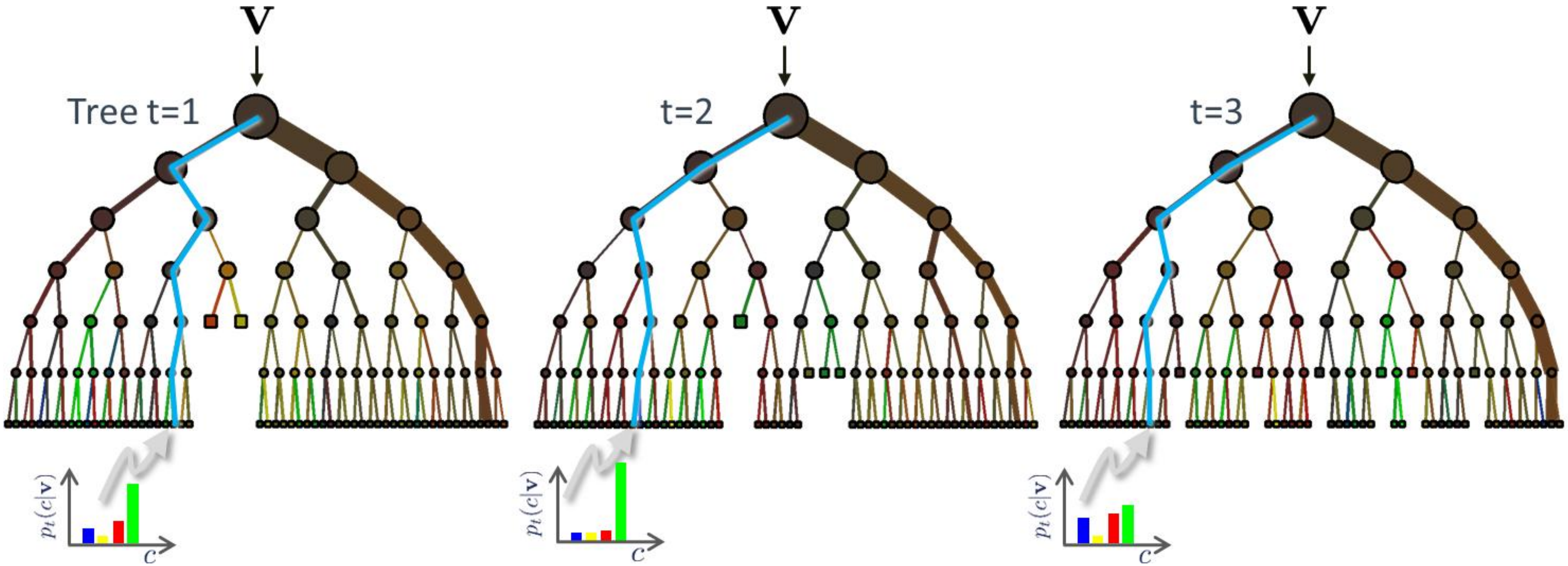
RANDOM FORESTS

Algorithm

- Draw a bootstrap sample from training data
- Grow a RF tree to the bootstrapped data:
  - Select m variables from p variables
  - Pick the best variable/splitting point among the m
  - Split the node into two child nodes
- Output ensemble of Trees

Building a Random Forest for Classification

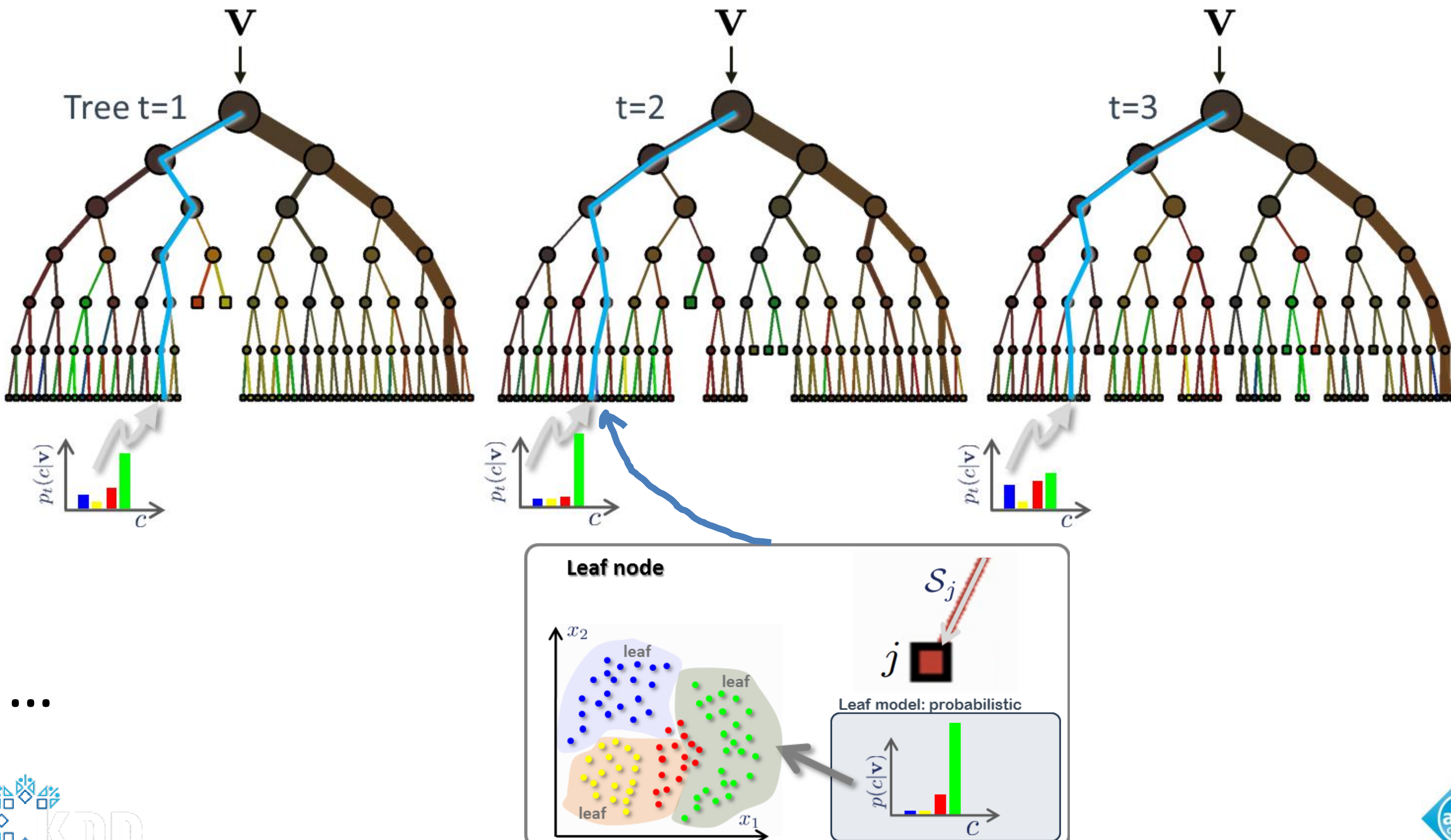
Problem Statement: *Given a labeled training set learn a general mapping which associates previously unseen test data with their correct classes.*



...

Building a Random Forest for Classification

Problem Statement: *Given a labeled training set learn a general mapping which associates previously unseen test data with their correct classes.*



…

Model Parameters

- Forest size (number of trees)
- Depth of Trees
- Amount of randomness and its type
  - Number of samples per node, criteria for selecting samples
- Choice of weak learner model
  - straight line, adaptive line, conical
- Training objective function
  - entropy, gini impurity
- Choice of features in practical applications
- Multiple Classes and Training Noise

→ ***Affect forest predictive accuracy and computational efficiency and…***

Random Forests

SCIKIT-LEARN COOKBOOK PYTHON CODE - DEMO

RandomForestClassifier(…)

- *class* sklearn.ensemble.RandomForestClassifier(*n_estimators=10*, *criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features='auto'*, *max_leaf_nodes=None*, *bootstrap=True*, *oob_score=False*, *n_jobs=1*, *random_state=None*, *verbose=0*, *warm_start=False*, *class_weight=None*)

- A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

RandomForestClassifier(...)

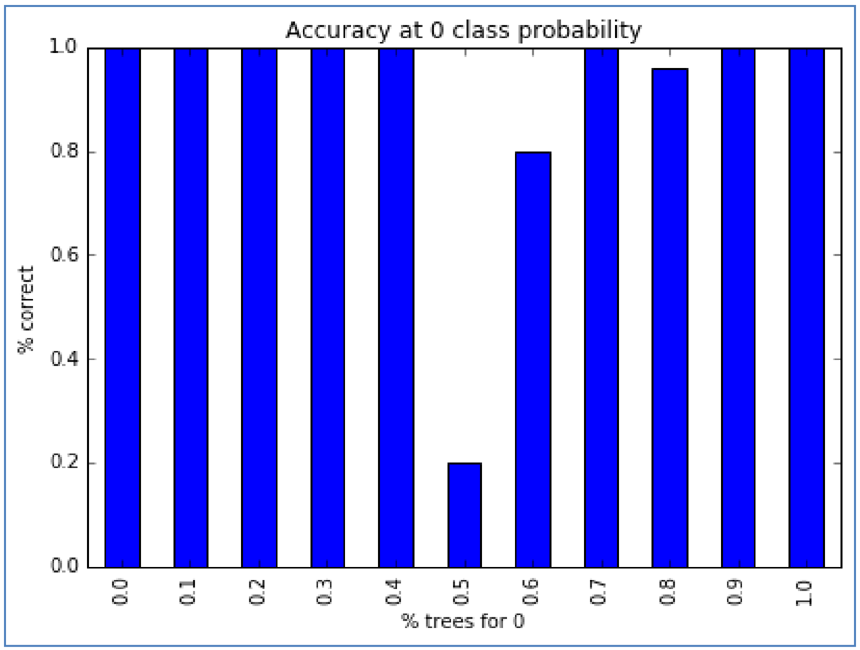
PARAMETERS	ATTRIBUTES	METHODS
n_estimators	estimators_	apply(X)
criterion	classes_	**fit(X, y)**
max_features	n_classes_	fit_transform(X[,y])
max_depth	**n_features_**	get_params([deep])
min_samples_split	**n_outputs_**	**predict(X)**
min_samples_leaf	**feature_importances_**	predict_log_proba(X)
min_weight_fraction_leaf	oob_score_	**predict_proba(X)**
max_leaf_nodes	oob_decision_function_	score(X, y[, sample_weisght])
bootstrap		set_params(**params)
oob_score		
n_jobs		
random_state		
verbose		
warm_start		
class_weight		
compute_importances		

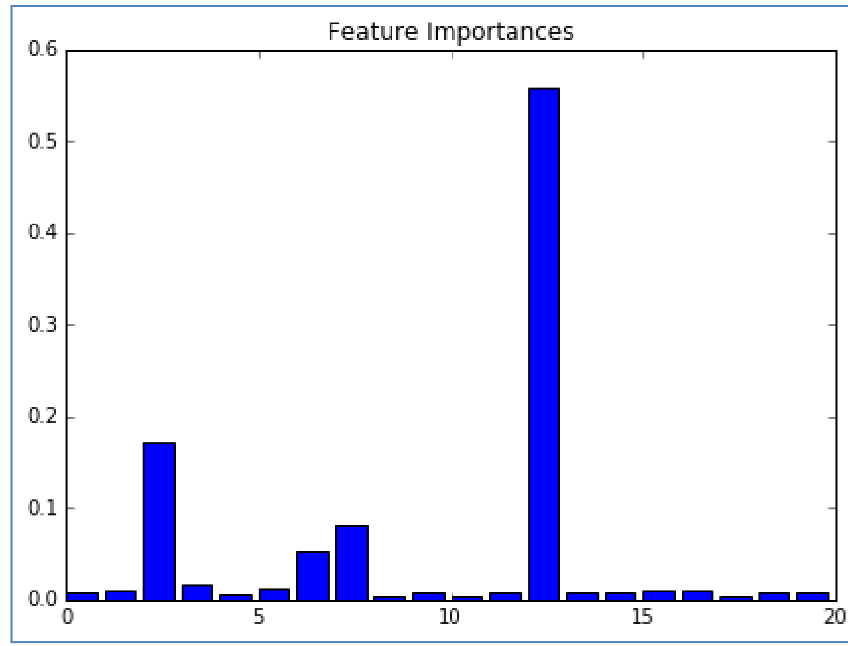
Code Review…DEMO

DEMO

Results of Text's code run: Random Forests

- Accuracy: 0.994
- Total Correct: 994



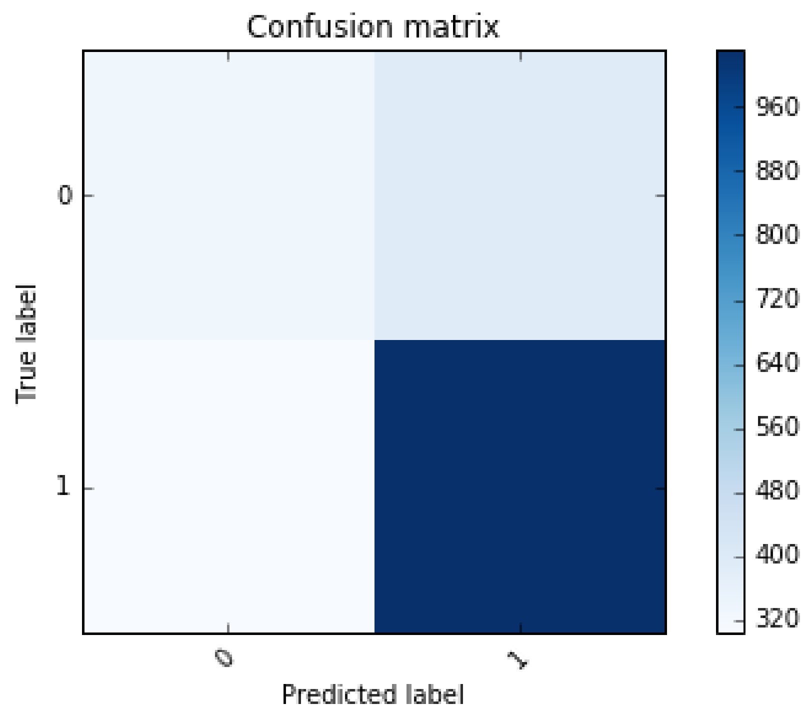


Model Evaluation: Accuracy, Confusion Matrix

- Accuracy: 0.661321671526
- Confusion Matrix for parameter value = auto number of features = 4

Values -
[[330 392]
[304 1032]]

Plot -

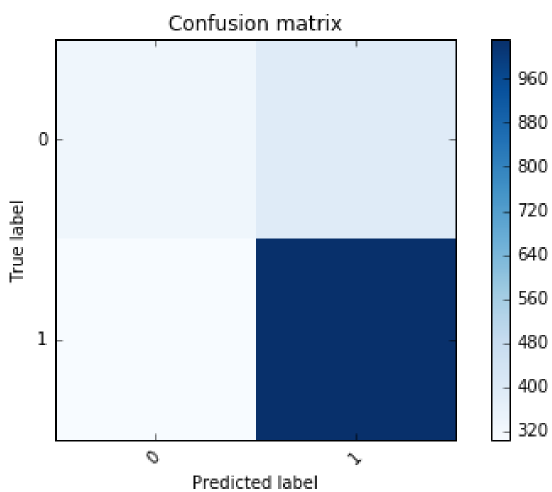


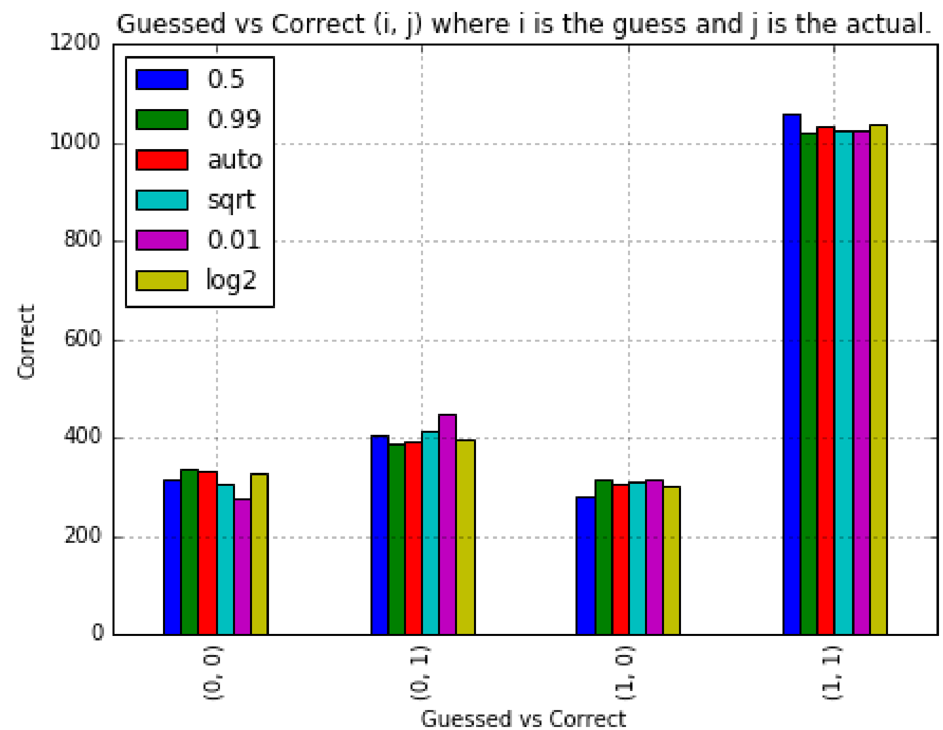
Model Evaluation: Confusion Matrix

- Confusion Matrix
  - For parameter value = ‘auto’, Number of features = 4
  - Values -

[[330 392]
 [304 1032]]

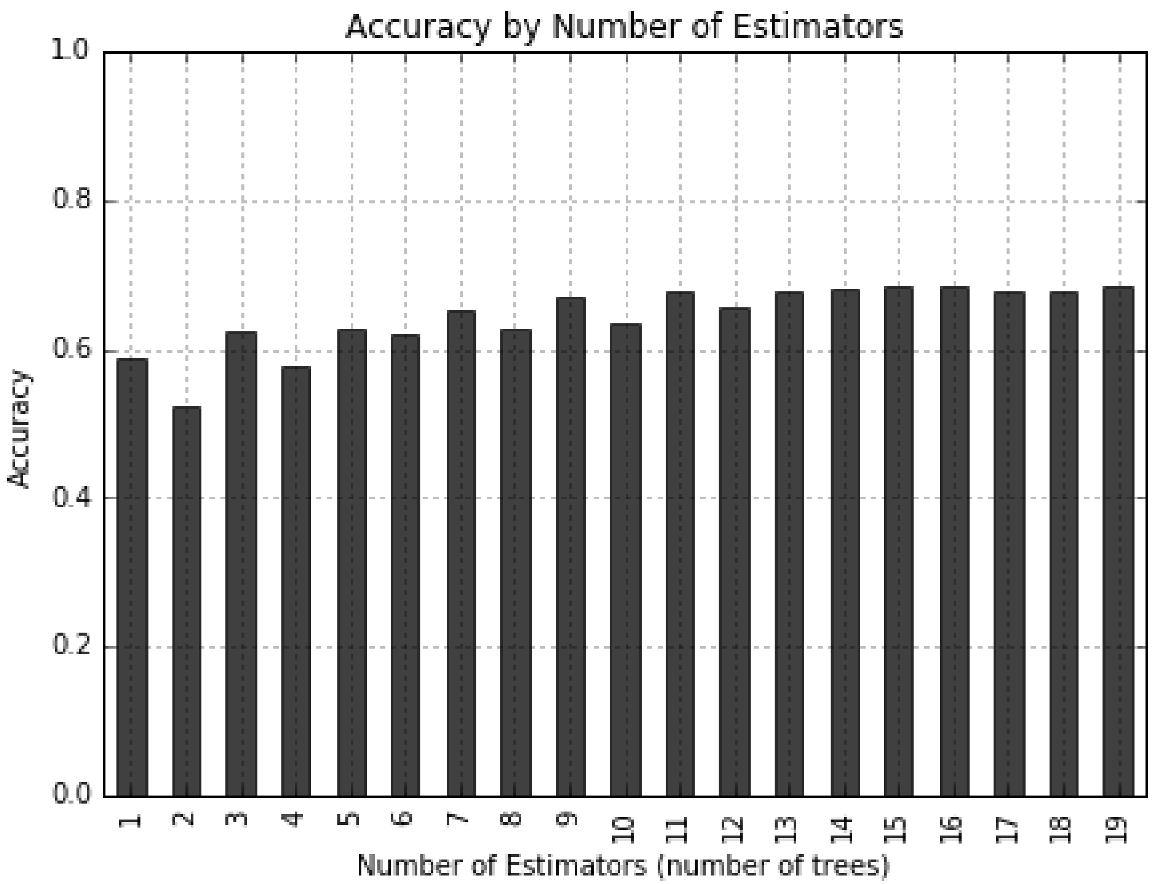
  - Plot -





Accuracy - by Number of Trees

Number of Trees = 20



Random Forests

SCIKIT-LEARN COOKBOOK PYTHON CODE - TUNING

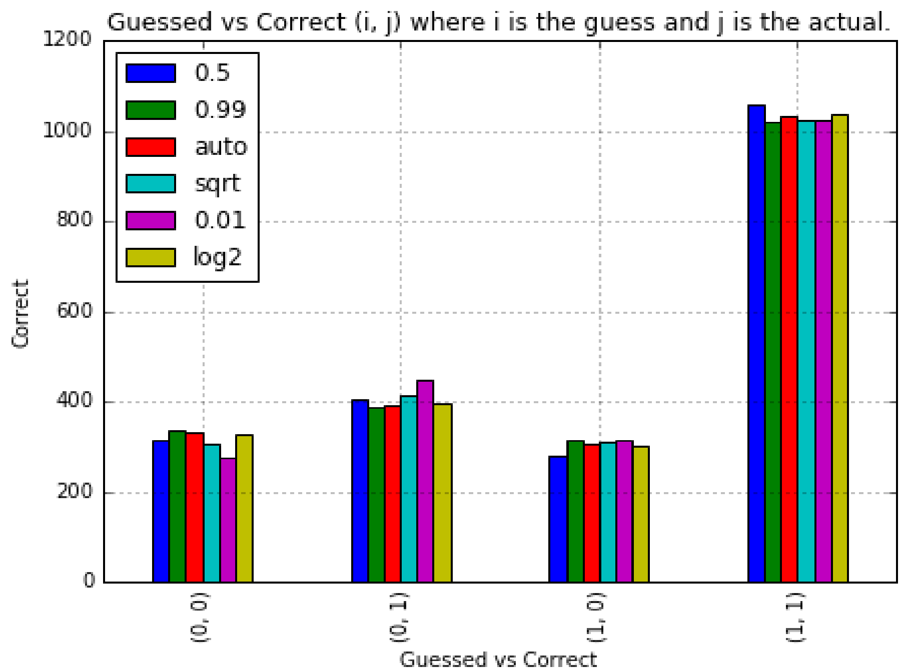
Tuning – Number of Features

Confusion Matrix for various number of features:

	0.5	0.99	auto	sqrt	0.01	log2
0	281	294	268	300	269	297
1	422	409	435	403	434	406
2	292	275	269	280	275	266
3	981	998	1004	993	998	1007

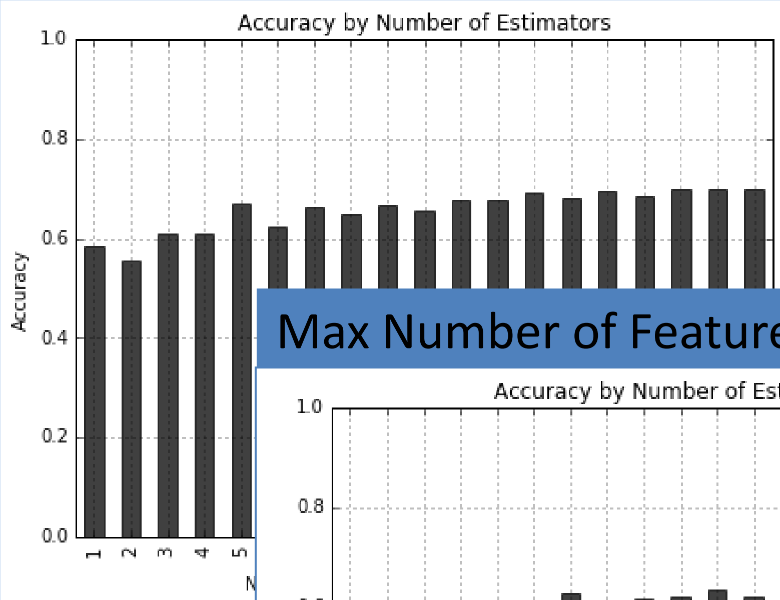
This is Random Forest Classifier with:

- **Number of Samples = 10000**
- **Number of Features = Varies**
- **Number of Trees = 20**
- **Depth of Trees = 2**
- **Node Splitting Criterion = 'gini'**
- **Number of cores = 1**

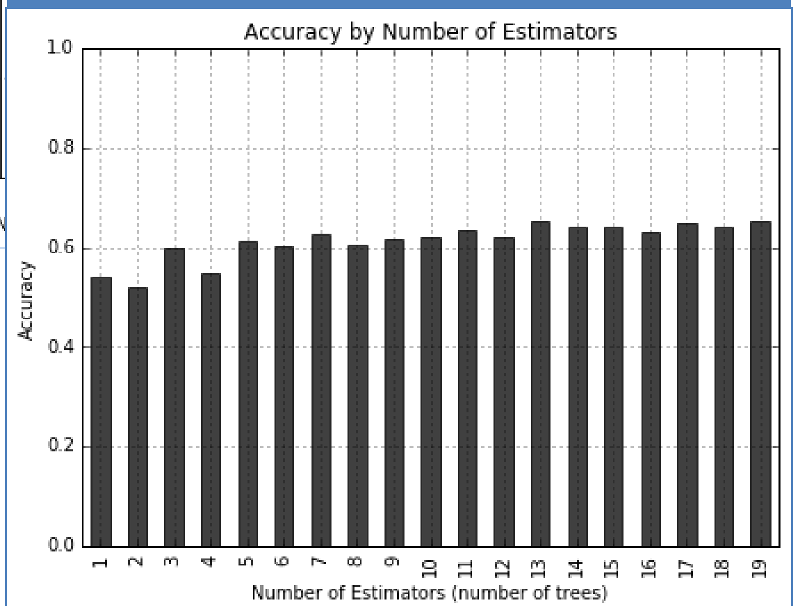


Tuning – Number of Features

Max Number of Features = 20



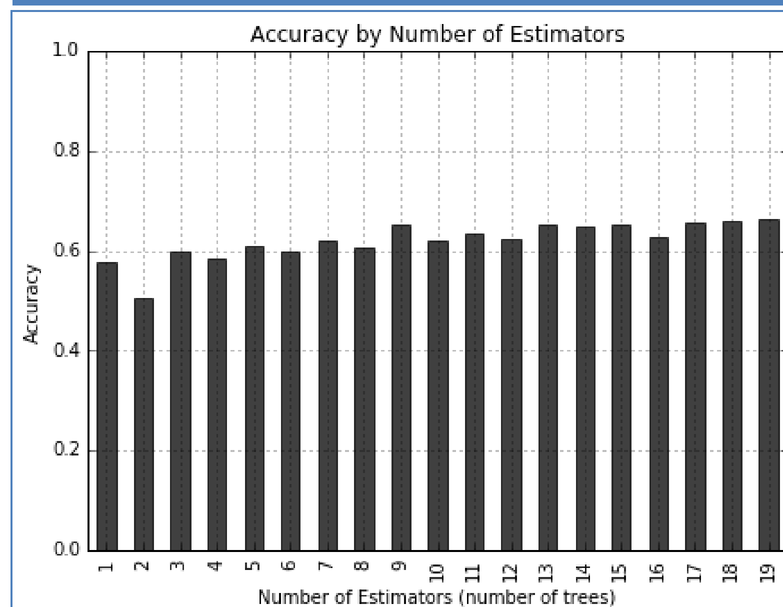
Max Number of Features = 50



This is Random Forest Classifier with:

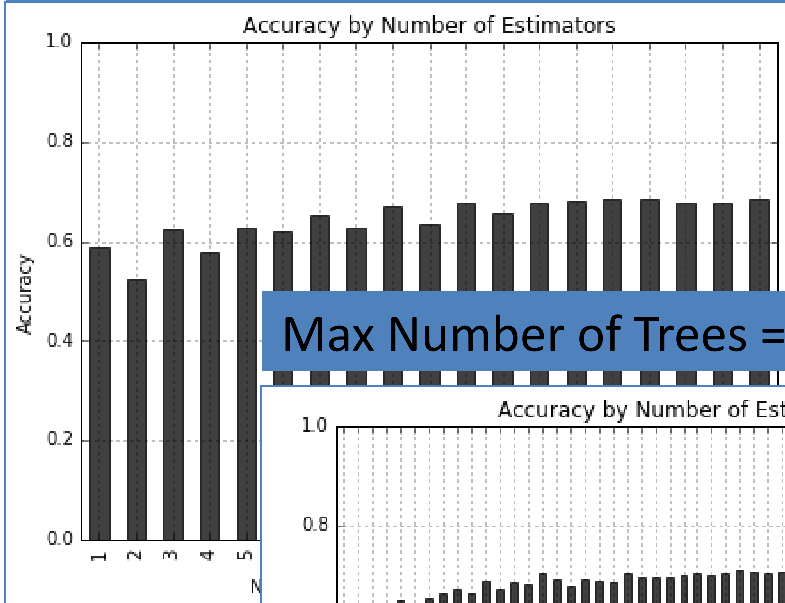
- **Number of Samples = 10000**
- **Number of Features = Varies**
- **Number of Trees = 1 to 20**
- **Depth of Trees = 2**
- **Node Splitting Criterion = 'gini'**
- **Number of cores = -1**

Max Number of Features = 100

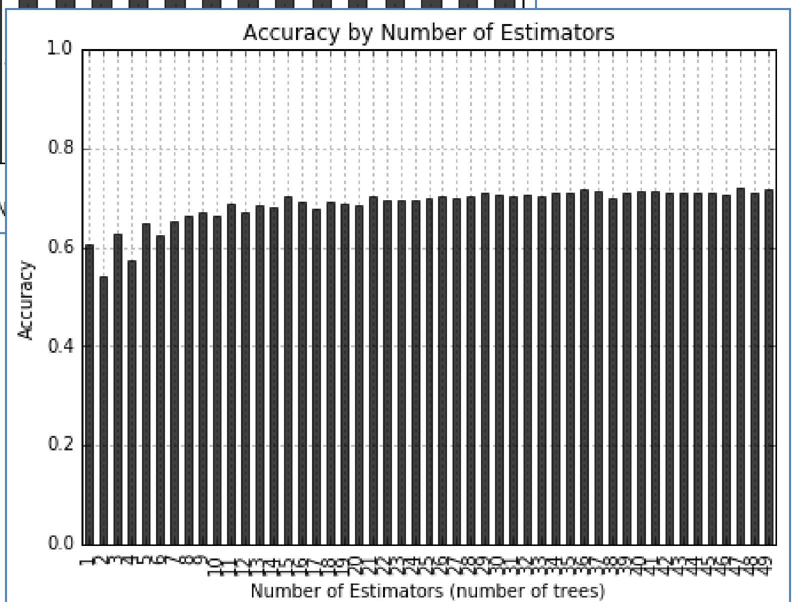


Tuning – Number of Trees

Max Number of Trees = 20



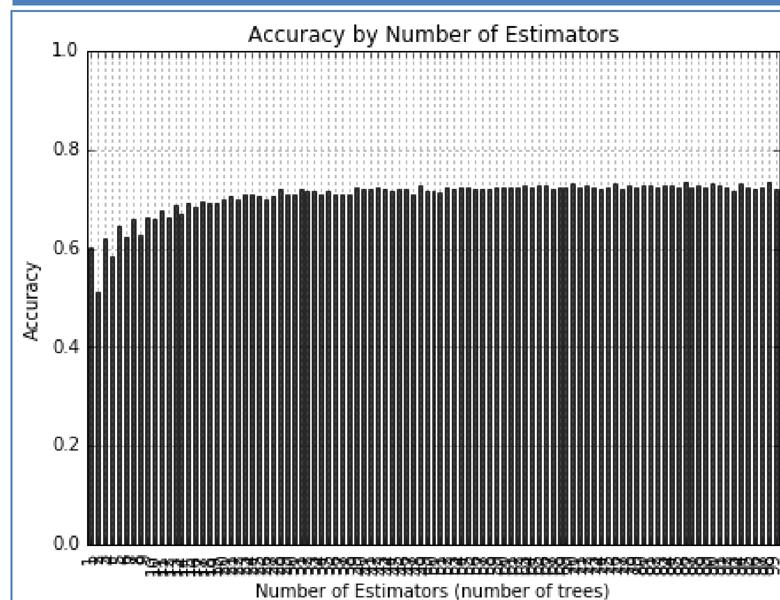
Max Number of Trees = 50



This is Random Forest Classifier with:

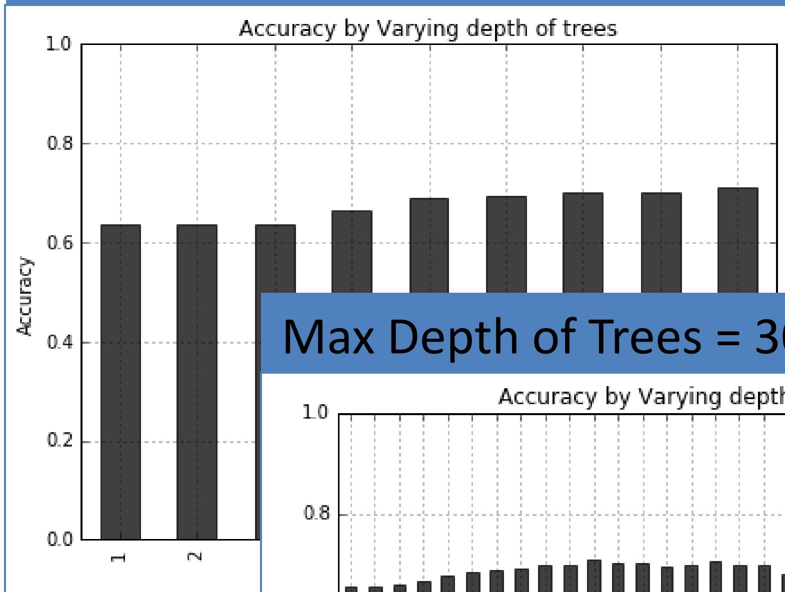
- **Number of Samples = 10000**
- **Number of Features = 20**
- **Number of Trees = Varies**
- **Depth of Trees = 2**
- **Node Splitting Criterion = 'gini'**
- **Number of cores = 1**

Max Number of Trees = 100

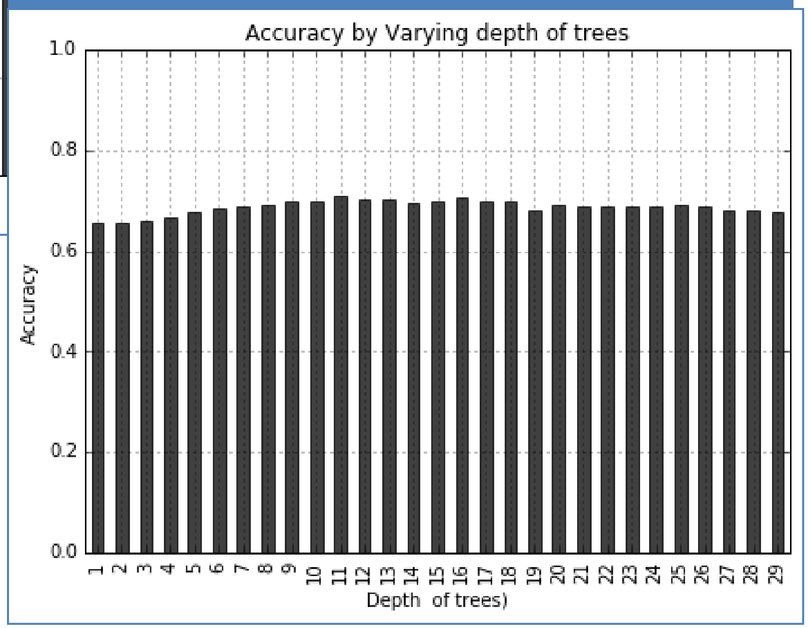


Tuning – Depth of Trees

Max Depth of Trees = 10



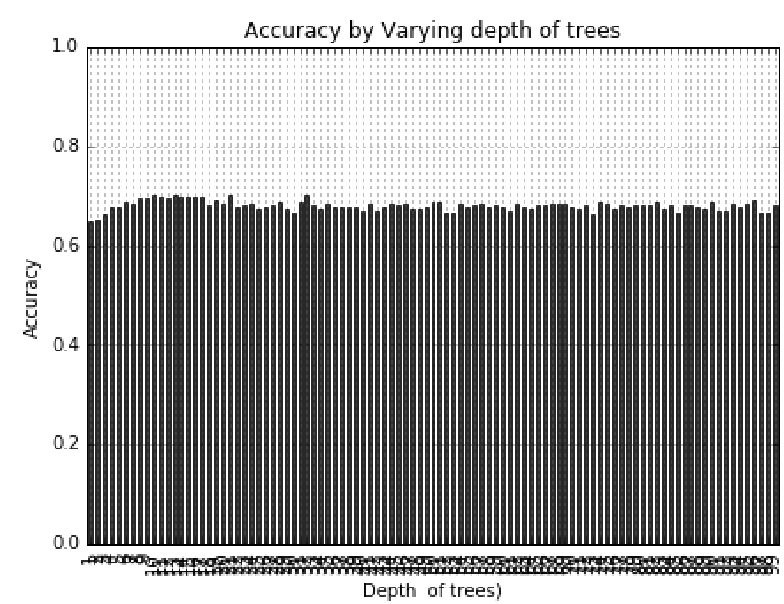
Max Depth of Trees = 30



This is Random Forest Classifier with:

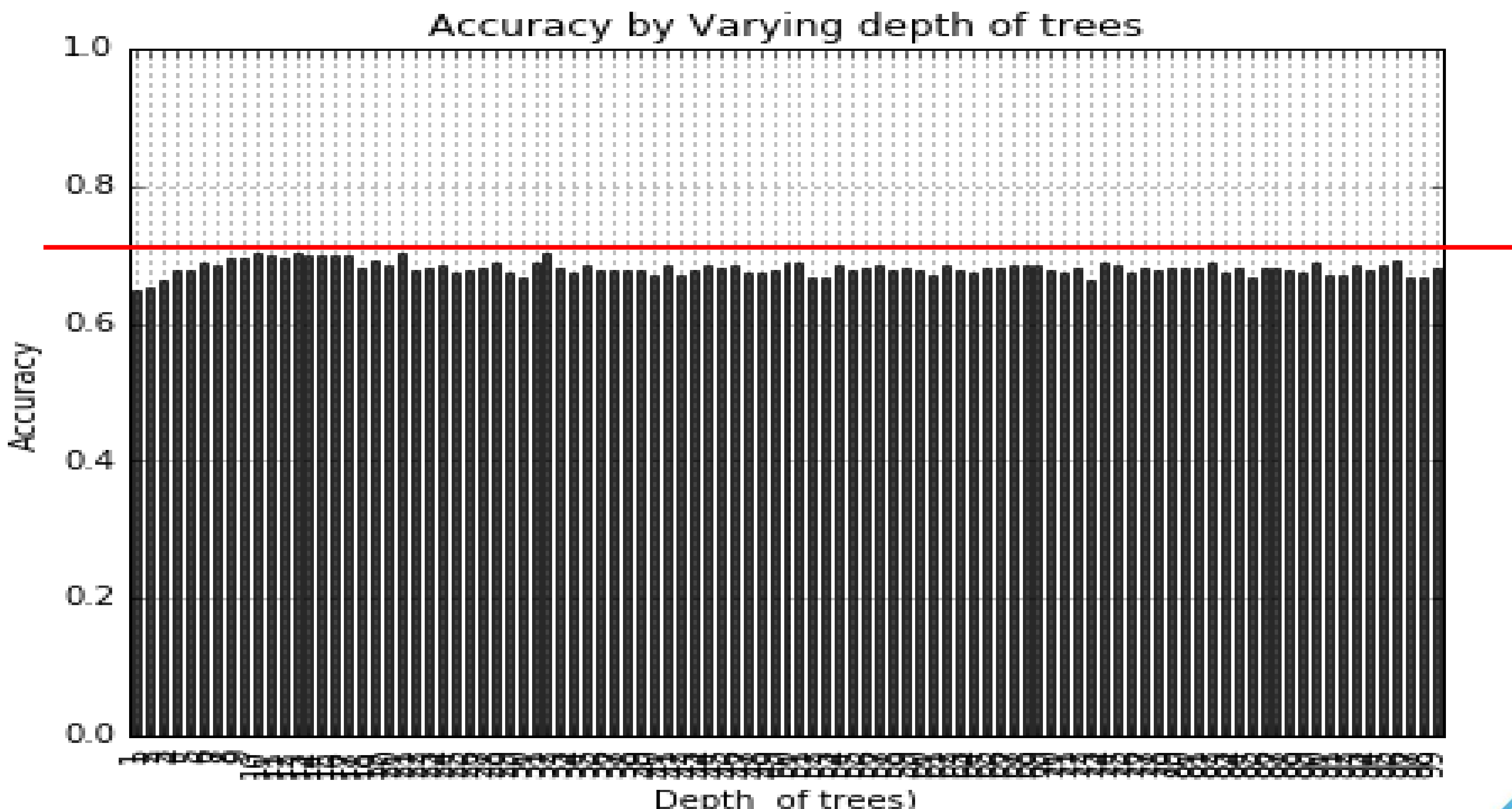
- **Number of Samples = 10000**
- **Number of Features = 20**
- **Number of Trees = 20**
- **Depth of Trees = Varies (…crazy)**
- **Node Splitting Criterion = ‘gini’**
- **Number of cores = 1**

Max Depth of Trees = 100



Tuning – Max Depth of Trees = 100 (crazy…)

Max Depth of Trees = 100



Tuning – Number of cores

This is Random Forest Classifier with:

- Number of Samples = 10000

- Number of Features = 20

- Number of Trees = 20

- Depth of Trees = 2

- Node Splitting Criterion = 'gini'

- Number of cores = Varies

My laptop is i5 (2 cores, 4 logical processors), Win 7

Number of Cores
Number of processors: -1 Time in seconds: 0:00:00.249615
Number of processors: 1 Time in seconds: 0:00:00.156009
Number of processors: 2 Time in seconds: 0:00:00.249615
Number of processors: 3 Time in seconds: 0:00:00.249614
Number of processors: 4 Time in seconds: 0:00:00.234014
Number of processors: 10 Time in seconds: 0:00:00.234008

Random Forests

APPLICATIONS

Applications of random decision forests

- Scene recognition in photos
- Object recognition in images
- Automatic diagnosis from radiological scans
- Semantic text parsing
- Text classification
- Face Detection
- Object Detection
- Kinect

References

- scikit-learn.org. Random Forests. http://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees

- Criminisi et all. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and Trends in Computer Graphics and Vision, 7(2-3): 81–227, 2011

- Denil et all: Narrowing the Gap - Random Forests In Theory and In Practice.

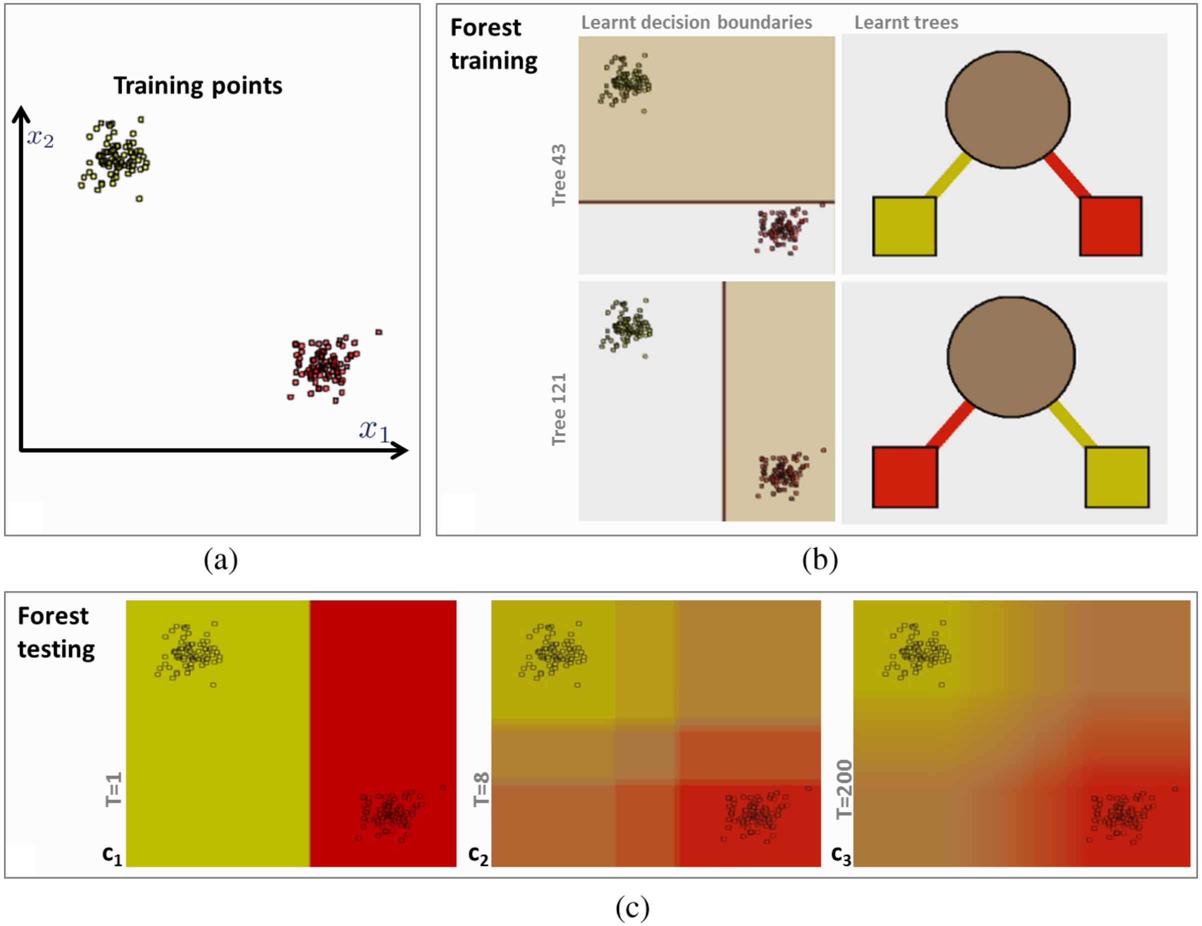
- Nando de Freitas . Machine learning - Random forests https://www.youtube.com/watch?v=3kYujfDgmNk

- Decision Tree Learning
  https://en.wikipedia.org/wiki/Decision_tree_learning

Random Forests

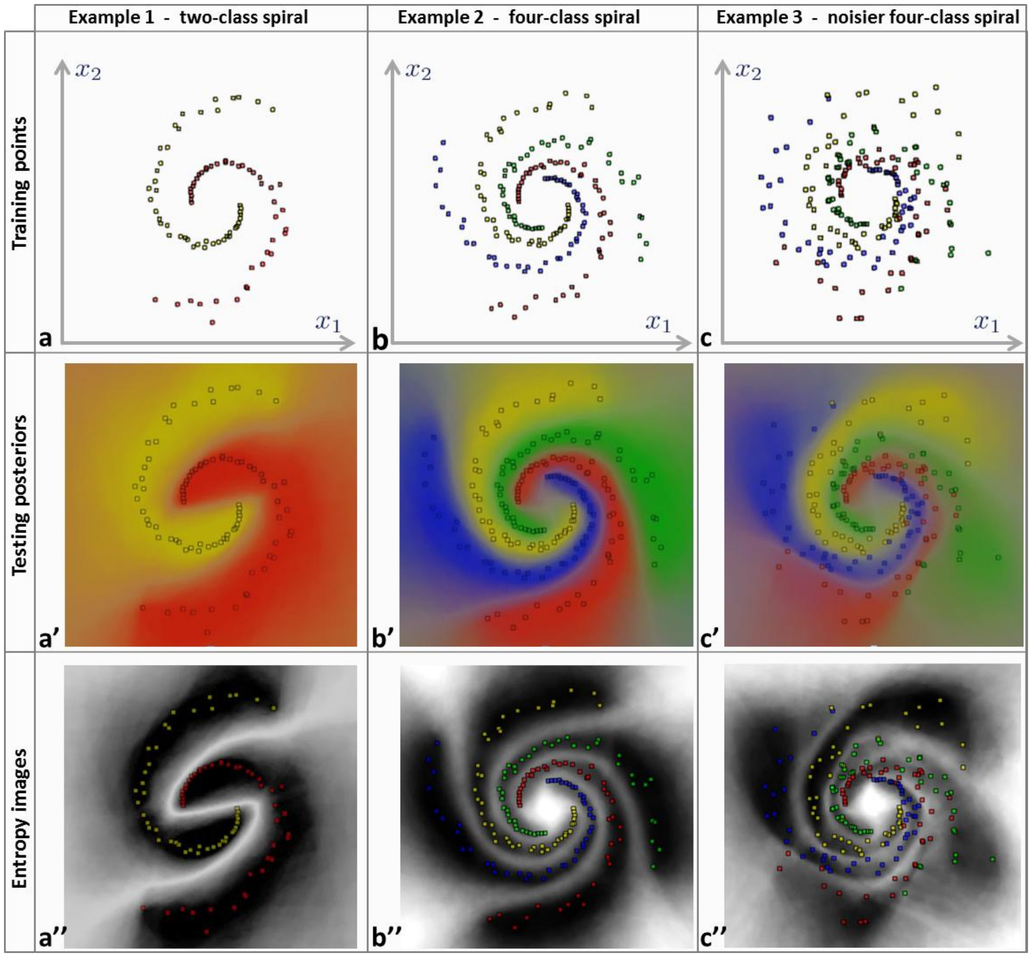
BACKUP

Model Parameter – Forest Size



- Increasing the forest size shows better results

Model Parameter – Multiple Classes and Training Noise



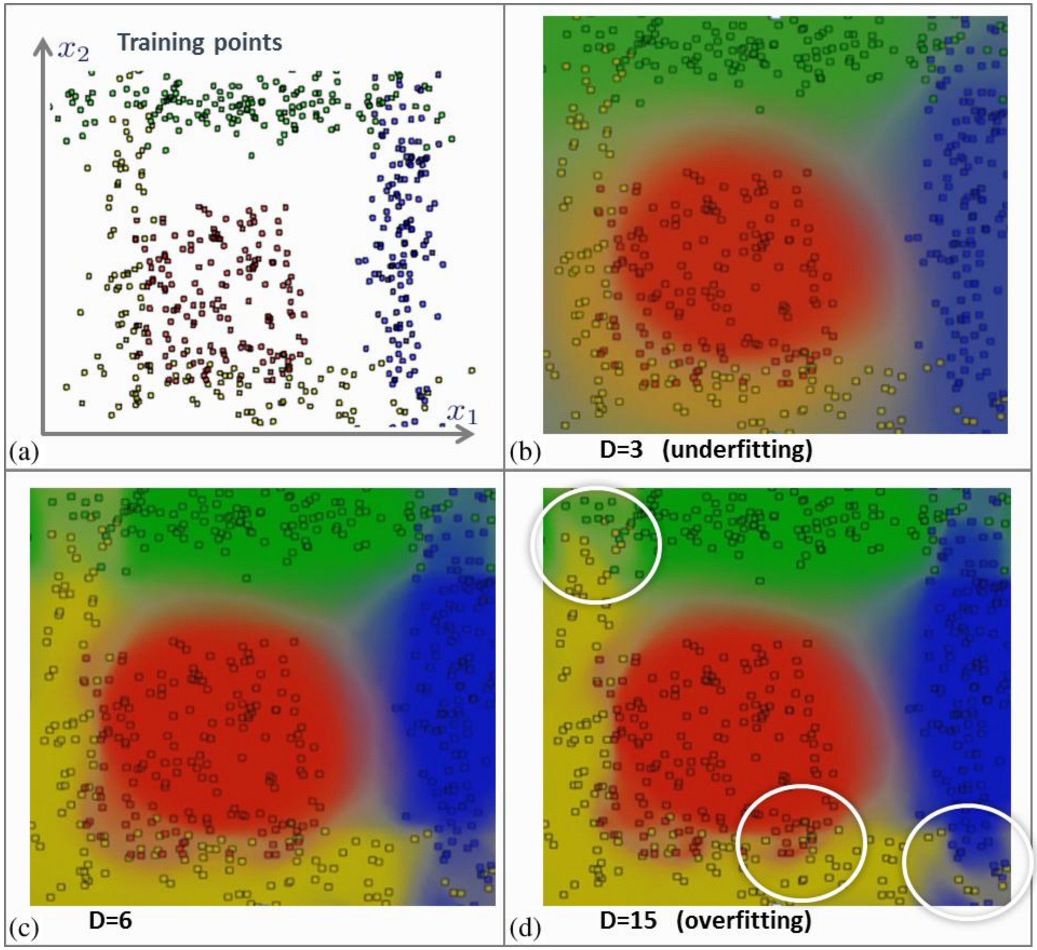
a, b, c: 2-class spiral, 4-class spiral and another 4-class spiral with noisier point positions, respectively

a',b',c': Corresponding testing posteriors.

a'', b'', c'': Corresponding entropy images (brighter for larger entropy).

- The classification forest can handle both binary as well as multi-class problems. With larger training noise the classification uncertainty increases (less saturated colors in c *and less sharp entropy in c).*

Model Parameter – Tree Depth

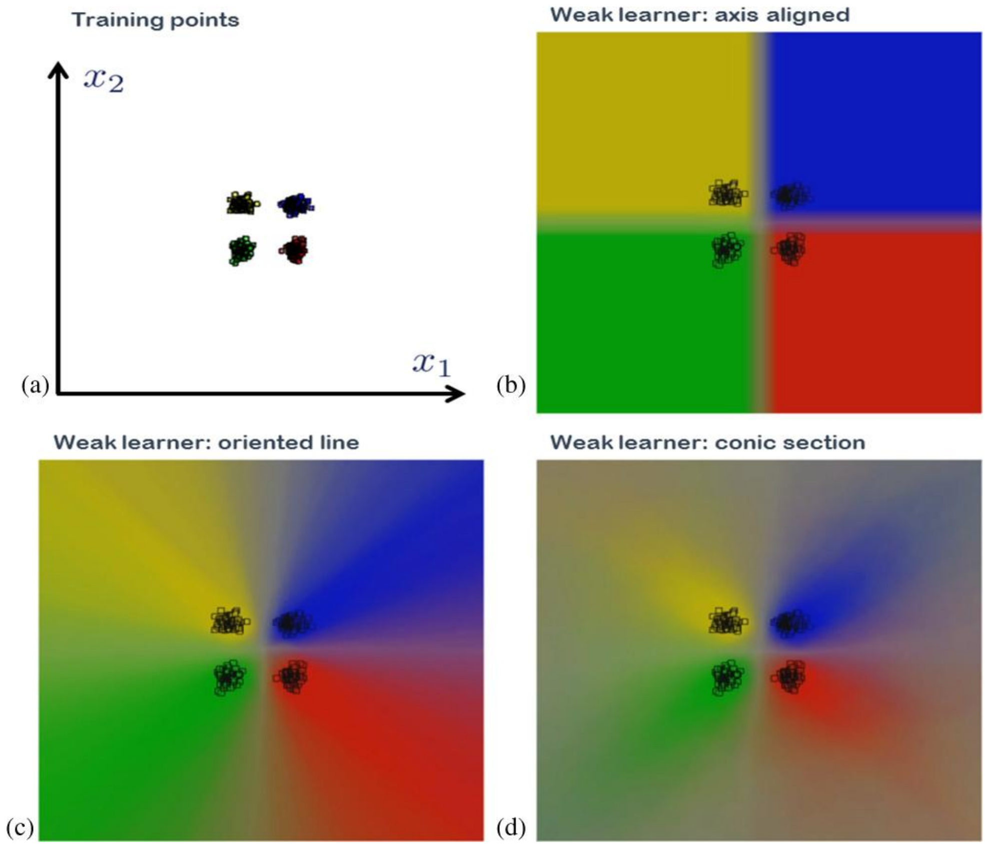


(a) Training points

(b,c,d) Testing posteriors for different tree depths.

- The tree depth is a crucial parameter in avoiding under- or over-fitting.

Model Parameter – Weak Learner



a) A four-class training set.
b) The testing posterior for a forest with axis-aligned weak learners. In regions far from the training points the posterior is overconfident.
c) The testing posterior for a forest with oriented line weak learners.
d) The testing posterior for a forest with conic section weak learners.

- Increasing *D increases the confidence* of the output (for fixed weak learner)
- Also consider the fact that axis aligned tests are extremely efficient to compute.
- So, the choice of the specific weak learner has to be based on considerations of both accuracy and efficiency and depends on the specific application at hand.

Impurity Measures

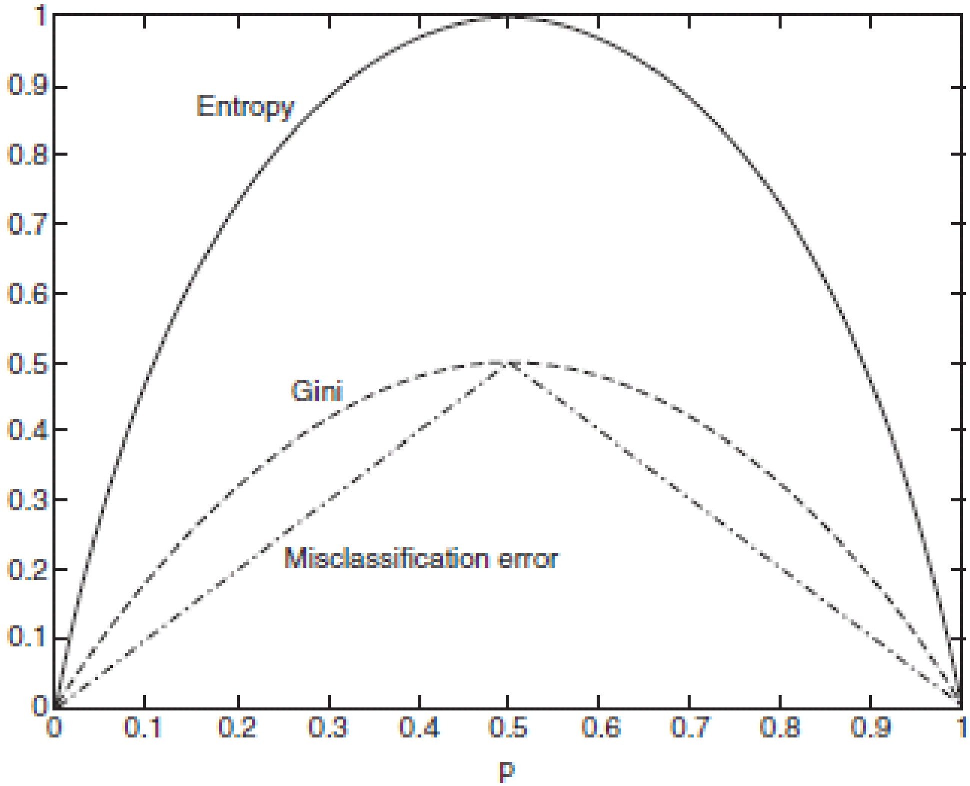
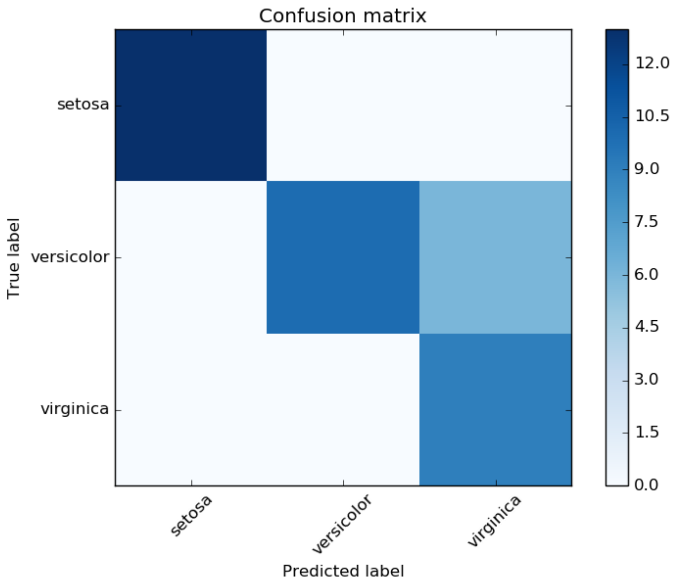


Figure 4.13. Comparison among the impurity measures for binary classification problems.

Reference: https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

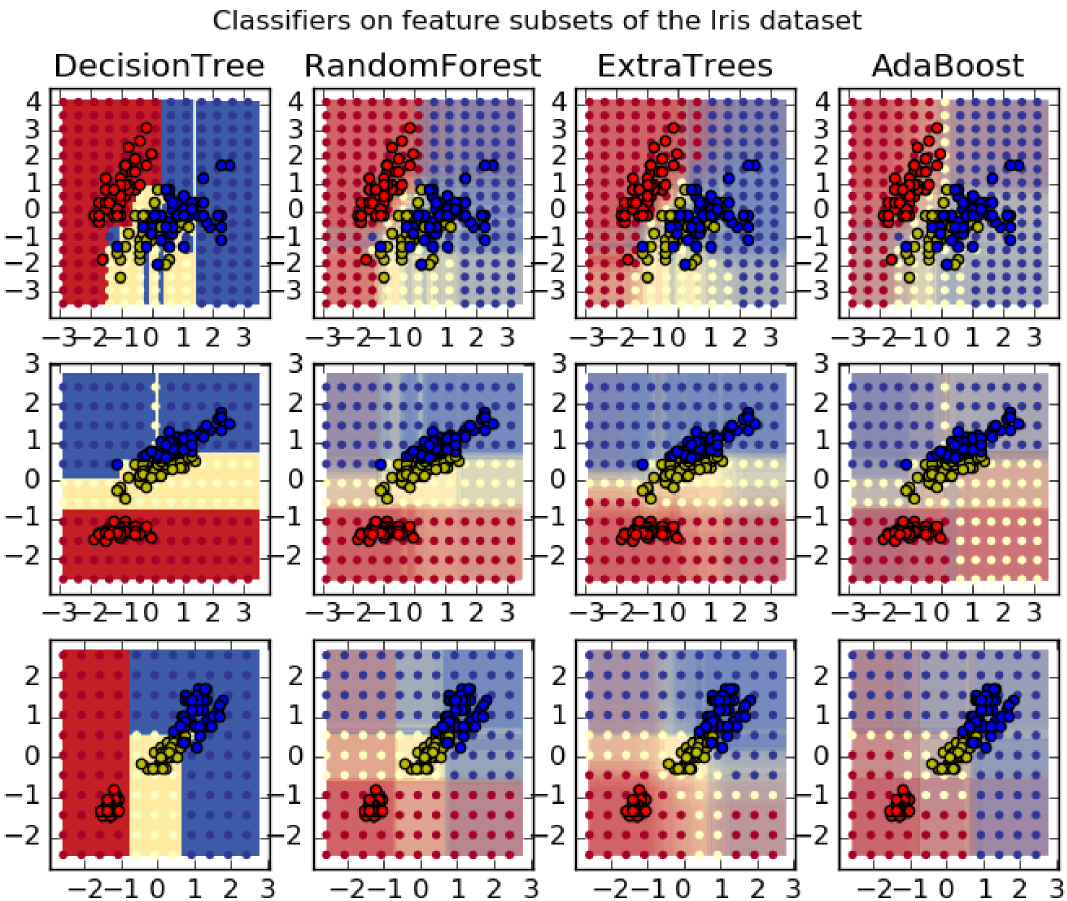
Confusion Matrix



http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

…

Decision surfaces of ensembles of trees on the iris dataset



http://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_iris.html#example-ensemble-plot-forest-iris-py

…